REMARKS

This amendment is responsive to the Office Action dated May 3, 2006. Applicant has amended claims 1, 14, and 27. Claims 1-30 are pending upon entry of this amendment.

Claim Rejection Under 35 U.S.C. § 103

In the Office Action, the Examiner rejected claims 1-30 under 35 U.S.C. 103(a) as being unpatentable over Oberman et al. (US 2001/0054140 A1) in view of Talwar et al. (US 2004/0230949 A1). Applicant respectfully traverses the rejection to the extent such rejections may be considered applicable to the claims as amended. The applied references fail to disclose or suggest the inventions defined by Applicant's amended claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

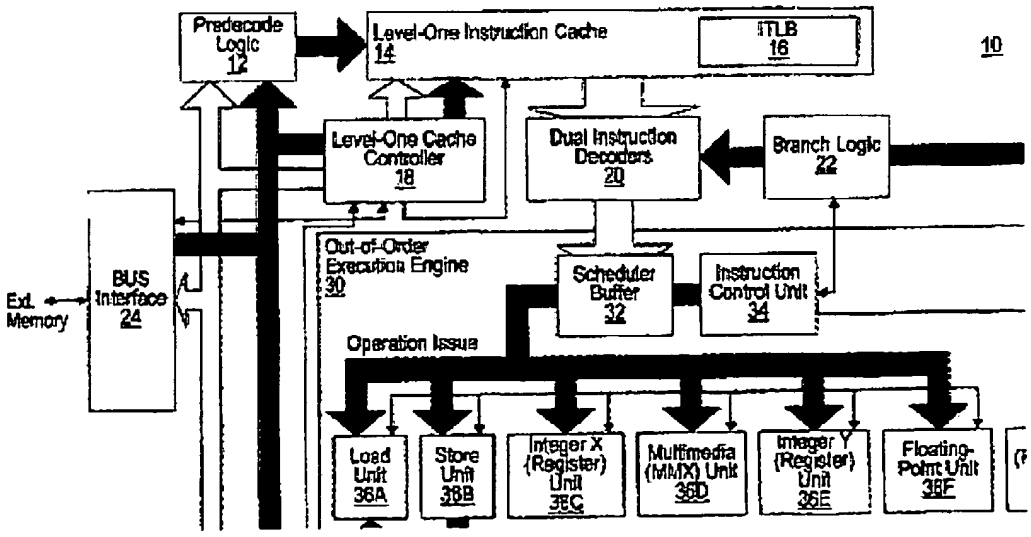
As a general comment to aid the Examiner's understanding, Applicant's claims are directed to emulation of a particular type of microprocessor, i.e., a microprocessor in which an instruction processor has separate interfaces to fetch op-codes and operands. That is, the microprocessor has separate interfaces by which op-codes and operands can be independently fetched. The claims are directed to embodiments for emulating such processors, which may be useful to test and verify the separate interfaces and the memory structures to which they connect. As discussed in further detail below, the primary reference, Oberman, describes a typical processor in which a microprocessor issues read request to a single interface to fetch entire instructions (i.e., opcodes and operands). Instructions are fetched as a whole and not by independently fetching opcodes and operands.

With reference to amended independent claim 1, the applied references lack any teaching that would have suggested a processor-based method performed by software emulating an instruction processor, where the method comprises processing read instructions with an emulated processor executing within an emulation environment to output independent read requests via an operand interface and an op-code interface of the emulated processor to independently fetch op-codes and operands from an emulated memory external from the emulated processor. The applied references also fail to teach or suggest the method further comprising independently comparing op-code reference data and operand reference data to operands and op-codes received

in response to the read requests and recording results of the independent comparisons. These arguments are also substantially applicable to independent claims 14 and 27 as amended.

The Examiner primarily relies on Oberman et al. ("Oberman") to reject the Applicant's claimed invention. In rejecting Applicant's claims, the Examiner characterizes Oberman as teaching processing read instructions with a processor to output independent read requests via an operand interface and an op-code interface. For support, the Examiner refers to Abstract lines 1-3 of Oberman, which state that "an execution unit is provided for executing a first instruction which includes an opcode field, a first operand field, and a second operand field." However, this passage refers to an execution unit within the Oberman microprocessor, and only states that the execution unit operates on instructions having opcodes and operands.

The Oberman execution unit 36D is internal to microprocessor 10, and operates on a decoded opcode value, as well as, first and second operand values from other components internal to the microprocessor, such as decode unit 20 and floating point unit 36F. Paragraph [0077], lines 1-3 and FIGS. 1. However, as shown in the reproduced portion of FIG. 1, microprocessor 10 fetches the instructions (including the opcode and the operands) from an external memory via a single BUS interface 24 and loads those instructions into level-one instruction cache 14 (Paragraph [0028]):



Oberman specifically states that the instruction fetch logic within cache controller 18 is capable of retrieving any 16 contiguous instruction bytes within a 32-byte boundary of cache 14. This makes it fairly clear that Oberman contemplates fetching contiguous instructions from cache 14 and TLB 16, and does not contemplate separate interfaces for independently fetching opcodes and operands. That is, based on FIG. 1 and the description provided by Oberman, it is clear that Oberman describes a typical microprocessor 10 where instructions are fetch whole using a single interface (either bus interface 10 or from level one instruction cache 14). Microprocessor 10 includes a single BUS interface 24 or a single interface to cache 14, as shown in the above reproduced portion of FIG. 1, to fetch instructions. Obermann simply does not teach or suggest a microprocessor where operands and op-codes can be independently fetched from an external memory using separate interfaces.

To the contrary, Applicant's claim 1 as amended requires processing read instructions with an emulated processor executing within an emulation environment to output independent read requests via an operand interface and an op-code interface of the emulated processor to independently fetch op-codes and operands from an emulated memory external from the emulated processor. This specifically requires emulation of a processor having multiple interfaces, i.e., an op-code interface and an operand interface, via which the emulated processor receives an opcode and an operand from an external memory.

The second reference, Talwar et al. ("Talwar"), provides no teaching to overcome the above deficiencies. As stated by the Examiner, Talwar merely discloses an emulation environment for verifying native language code. Office Action, page 3.

For at least these reasons, Oberman in view of Talwar provide no teaching for emulating an instruction processor capable of outputting independent read requests via an operand interface and an op-code interface of the emulated processor to independently fetch op-codes and operands from an emulated memory external from the emulated processor, as required by Applicant's claim 1 as amended.

The applied references also fail to disclose the additional requirements of Applicant's dependent claims 2-13, 15-26, and 28-30. With respect to claim 2, for example, the applied references fail to disclose storing the op-code reference data and the operand reference data within a set of data memories of the emulated instruction processor, maintaining within the

emulated instruction processor an operand data pointer to address the operand reference data and an op-code pointer to address the op-code reference data, and independently accessing the operand reference data with the operand data pointer and the op-code reference data with the op-code data pointer during processing of the read instructions to verify the received op-codes and the received operands.

In part, the Examiner supports the rejection of claim 2 by again referencing Abstract lines 1-3 of Oberman. Lines 1-3 of the Abstract make no mention of nor could it be construed to suggest maintaining pointers to address op-code reference data and operand reference data. The Examiner provides no explanation as to why an execution unit for executing a first instruction which includes an opcode field, a first operand field, and a second operand field would necessarily maintain or even require pointers to address op-code reference data and operand reference data. The mere fact that the execution unit processes instructions having an opcode and two operands does not require that the execution unit maintain pointers to reference the opcode and the two operands. Moreover, the pointers disclosed in Applicant's claim 2 reference opcode reference data and operand reference data for comparison purposes and not the actual opcode data and operand data defined by an instruction.

For at least these reasons, the Examiner has failed to establish a prima facie case for non-patentability of Applicant's claims 1-30 under 35 U.S.C. 103(a). Applicant respectfully requests withdrawal of this rejection.

CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Application Number 10/729,666
Responsive to Office Action mailed May 3, 2006

Date: August 2, 2006

By: Kent J. Sieffert

SHUMAKER & SIEFFERT, P.A.
8425 Seasons Parkway, Suite 105
St. Paul, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102

Name: Kent J. Sieffert
Reg. No.: 41,312